remote sensing

MDPI

Article

Loop Detection Method Based on Neural Radiance Field BoW Model for Visual Inertial Navigation of UAVs

Xiaoyue Zhang [1,2], Yue Cui [1,2], Yanchao Ren [3,*], Guodong Duan [3] and Huanrui Zhang [1,2]

[1] School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China; zhangxiaoyue@buaa.edu.cn (X.Z.); zy2217111@buaa.edu.cn (Y.C.); zhanghr@buaa.edu.cn (H.Z.)
[2] The National Key Laboratory of Inertial Technology, Beihang University, Beijing 100083, China
[3] Hunan Vanguard Group Company Limited, Changsha 410137, China; dgdguo@163.com
* Correspondence: hanchen715@163.com

Abstract: The loop closure detection (LCD) methods in Unmanned Aerial Vehicle (UAV) Visual Inertial Navigation System (VINS) are often affected by issues such as insufficient image texture information and limited observational perspectives, resulting in constrained UAV positioning accuracy and reduced capability to perform complex tasks. This study proposes a Bag-of-Words (BoW) LCD method based on Neural Radiance Field (NeRF), which estimates camera poses from existing images and achieves rapid scene reconstruction through NeRF. A method is designed to select virtual viewpoints and render images along the flight trajectory using a specific sampling approach to expand the limited observational angles, mitigating the impact of image blur and insufficient texture information at specific viewpoints while enlarging the loop closure candidate frames to improve the accuracy and success rate of LCD. Additionally, a BoW vector construction method that incorporates the importance of similar visual words and an adapted virtual image filtering and comprehensive scoring calculation method are designed to determine loop closures. Applied to VINS-Mono and ORB-SLAM3, and compared with the advanced BoW model LCDs of the two systems, results indicate that the NeRF-based BoW LCD method can detect more than 48% additional accurate loop closures, while the system's navigation positioning error mean is reduced by over 46%, validating the effectiveness and superiority of the proposed method and demonstrating its significant importance for improving the navigation accuracy of VINS.

Keywords: NeRF; VINS; Bag-of-Words; loop closure detection

Citation: Zhang, X.; Cui, Y.; Ren, Y.; Duan, G.; Zhang, H. Loop Detection Method Based on Neural Radiance Field BoW Model for Visual Inertial Navigation of UAVs. *Remote Sens.* **2024**, *16*, 3038. https://doi.org/10.3390/rs16163038

Academic Editor: Andrzej Stateczny

Received: 7 June 2024
Revised: 30 July 2024
Accepted: 18 August 2024
Published: 19 August 2024

1. Introduction

Unmanned Aerial Vehicles (UAVs) are currently being used in a wide range of applications. The Visual Inertial Navigation System (VINS) has become the dominant method for enabling autonomous navigation of UAVs. VINS combines input from visual sensors with Inertial Measurement Units (IMUs) to calculate the position, velocity, and attitude of a UAV in real time. This allows the UAV to quickly adjust to different situations and effectively complete its navigation objectives [1,2]; achieving high-precision navigation positioning serves as a fundamental requirement for UAVs to accomplish complex tasks. However, when employing Visual Inertial Odometry (VIO) for navigation, cumulative errors may arise, which can degrade navigation accuracy; error accumulation can impede their applications across different domains [3]. Loop closure detection (LCD) approaches can significantly reduce pose drift errors accumulated by VIO and improve navigation accuracy by determining if the system has revisited previously traversed places and implementing appropriate correction measures [4]. Presently, commonly employed LCD techniques often entail extracting features from images and seeking correspondences between pairs of images. These approaches can be broadly classified into two categories: those that rely on deep learning and those that rely on Bag-of-Words (BoW) models. This

paper uses the BoW model as the framework for designing the method, primarily for the following four reasons. Firstly, UAVs have a wide range of applications and sometimes lack the opportunity for pre-training before executing their tasks. This necessitates a flexible and readily deployable approach. Secondly, our research focuses on small UAVs, which have limited capacity to carry heavy computational equipment. Integrating high-performance computational devices into small UAVs poses significant challenges due to their size and weight constraints. Thirdly, the BoW model offers simplicity in implementation. It does not require the complex network training process inherent in many deep learning methods, making the development and debugging process more straightforward. This simplicity ensures that the system can be efficiently developed and maintained. Finally, deep learning models frequently exhibit a black-box nature, complicating the debugging process. Problems can originate from various sources, such as data quality, model architecture, or training procedures, and these issues often require sophisticated tools and extensive experiments to diagnose and resolve. Consequently, the BoW model provides a more transparent and manageable alternative, suitable for the computational and operational constraints of small UAVs.

The BoW concept was initially developed in the field of natural language processing [5] and subsequently applied to the domain of VINS. BoW model-based LCD determines loop closures by comparing the consistency of "words" in two images, expressing images as vectors, and calculating similarity scores between images based on vector norms [6]. Currently, BoW model-based LCD is widely adopted by mainstream open-source VINS, resulting in a significant enhancement in detection performance and system localization accuracy [7,8]. The key to BoW model-based LCD lies in extracting suitable feature points to achieve better clustering and matching. However, in areas with low density of texture, the extractable feature points may significantly decrease or even vanish [9]. In application scenarios of UAVs, the high speed of aircraft movement makes it difficult to track, extract, and match feature points in scenes with certain lighting changes and weak textures. Furthermore, the restricted number of observation viewpoints offered by the flight path amplifies the challenge of identifying loop closures. Previous efforts to address this issue through scene reconstruction have often been impeded by slow reconstruction speeds. This significantly limits the error correction capabilities of the LCD process, making it impractical for real-world applications [10]. This study proposes using Neural Radiance Field (NeRF) for rapid [11], high-quality scene reconstruction, expanding image sequences by generating novel views to increase the number and optimize the quality of available images, thereby enhancing the detection rate of LCD.

This study proposed a BoW model based the LCD method utilizing NeRF, which is a type of neural rendering technique developed in recent years to overcome the performance limitations of traditional 3D reconstruction methods by employing differentiable rendering and neural networks. NeRF proposes using neural implicit fields for continuous scene representation and has achieved significant success in synthesizing high-quality views and 3D reconstruction in various scenes. Its core idea involves employing a multilayer perceptron network to learn volumetric information of the 3D scene from input 2D images. The advantages of this technique lie in its ability to generate high-quality 3D reconstructions and its efficiency and flexibility compared to traditional 3D modeling methods. Currently, NeRF has demonstrated broad potential applications in multiple domains. In computer vision, NeRF is utilized for high-precision 3D scene reconstruction and novel view synthesis [12]. In virtual and augmented reality, NeRF enhances immersive experiences [13]. In film and game production, NeRF generates high-quality visual effects and scenes [14]. With technological advancements, NeRF has also been applied in fields such as robot navigation, with some research focusing on integrating NeRF within Simultaneous Localization And Mapping (SLAM) systems [15], and others using pre-trained NeRF maps for localization and optimizing vehicle trajectories [16]. Leveraging NeRF's view synthesis capabilities, this study proposes the following contributions:

1. Adopting the rapid neural radiance field Instant Neural Graphics Primitives (Instant-NGP) [17] as a scene reconstruction tool, we propose a method that utilizes reconstructed scenes to obtain virtual viewpoint images near the flight trajectory through a specific sampling approach. This method aims to increase the number of observation angles and expand the loop closure candidate images. By providing more diverse scene information, the proposed method enhances the success rate and accuracy of LCD.
2. Designed a similarity factor-based method to construct BoW vectors with word frequency weight, which utilizes cosine similarity and dynamic weight assignment to obtain comprehensive similarity scores for loop closure determination. Specifically, the similarity between new words in virtual images and existing words is considered to reduce the probability of false positives in loop detection caused by the introduction of virtual images, thereby preventing the system from making incorrect corrections based on erroneous loop information.

The second part of this paper introduces the existing LCD techniques in VINS, along with their advantages and limitations. It also provides an overview of the current development status and superiority of NeRF, which forms the basis of the suggested approach. Section 3 provides a detailed explanation of the BoW model-based LCD approach using NeRF. It specifically focuses on the process of selecting virtual view poses and constructing word frequency weight vectors. Section 4 of the document outlines the experimental procedure and presents the findings, which clearly illustrate the exceptional effectiveness of the suggested strategy in dynamic environments. The fifth part discusses the results of the method design in the context of VINS research, summarizes the strengths and weaknesses of the proposed method, and proposes further research directions.

2. Related Work

Deep learning-based LCD approaches generally demonstrate higher levels of robustness in complicated situations. Nevertheless, the current mainstream LCD approaches in VINS depend heavily on BoW models due to constraints in real-time performance and processing capacity. Both approaches encounter difficulties in obtaining characteristic points when there are constraints on the observer viewpoints, low-textured surfaces, or variations in illumination conditions. NeRF's capacity to generate perspectives and effectively recreate scenes provides methods to tackle these problems.

2.1. Deep Learning-Based Loop Closure Detection Methods

The application of deep learning models such as Convolutional Neural Networks (CNNs) and autoencoders in LCD has garnered significant attention, prompting numerous related attempts by researchers.

Chen attempted to apply CNN to a location recognition dataset spanning 70 km [18], constructing a confusion matrix for matching. Hou compared the performance of CNN-extracted features with traditional descriptors in loop detection [19], finding that CNN features performed better when the operating environment experienced lighting changes. Ma proposed the Local Relative Orientation (LRO) matching algorithm to compute correspondences between image pairs [20], demonstrating significant robustness in scenarios with viewpoint changes and dynamic objects. Hao used ResNet to extract global image features and combined sequence image features as the features for the current frame [21], which is more suitable for large-scale scenes. Sunderhauf discovered that intermediate layer feature encoding in CNNs is robust to conditions like weather and lighting [22], while top layer feature encoding is robust to viewpoint changes.

Nevertheless, the characteristics obtained by algorithms based on CNN are usually of high dimensionality and require significant processing resources. As a result, many techniques for reducing the dimensionality are being explored. For example, Luo employed the T-distributed Stochastic Neighbor Embedding (TSNE) technique to decrease the number of dimensions in the high-dimensional features acquired from the Visual Geometry Group

16 (VGG16) network [23], hence removing redundant data. Sunderhauf employed a binary local sensitive hashing algorithm to decrease the dimensionality of picture information while preserving 95% of the location identification performance [22].

2.2. BoW Model-Based Loop Closure Detection Methods

BoW-based LCD methods treat image features as "words". Initially, keypoints are extracted from images and descriptors are generated using a feature extraction algorithm. Subsequently, clustering algorithms are employed to construct a bag of visual words from these descriptors, enabling vectorized representation of images and computation of similarity between images.

Lopez suggested a technique called the LCD method, which utilizes the FAST points and BRIEF descriptors [24]. This method employs a K-tree representation of the dictionary to enhance the speed of the search process. Nevertheless, it cannot maintain its accuracy and consistency when subjected to rotation and scale changes, rendering it inappropriate for applications using drones. Labbe presented RTAB-Map, a software package that incorporates a memory management-driven LCD algorithm [25]. This algorithm efficiently utilizes a restricted set of sites for LCD and systematically visits all locations as needed. Garcia implemented a hierarchical binary BoW for LCD [26] that was updated progressively. This approach allowed for efficient real-time search, insertion, and deletion of new visual words, resulting in improved real-time performance. Tsintotas suggested an incremental BoW model for LCD [5], which encodes traversed paths by utilizing a small number of distinct visual words obtained from the feature tracking procedure. Certain studies proposed the integration of point-line features into loop closure identification by proposing a Line Band Descriptor (LBD) and a data-dependent point-line feature-based LCD method [27]. This algorithm detects loops by considering data dependencies and calculating similarity.

BoW methods utilize vectors to represent images, calculate image similarity, and identify loops. These methods can be integrated with image sequence and semantic information to improve reliability. Deep learning-based methods construct visual descriptions using deep learning models, which provide superior accuracy and resilience. However, these methods are limited in their use due to constraints in device resource allocation and dataset needs. BoW-based approaches continue to be the prevailing approach, whereas deep learning-based methods are still in a phase of development and experimentation.

2.3. Neural Radiance Fields

NeRFs have gained significant traction in recent years, as they offer a solution to the challenge of representing 3D scenes without the need for extensive storage capacity. As a unique and widely accepted technique for representing scenes, it has been successful in generating new views of scenes [11,28–30]. The majority of NeRF works operate under the assumption that camera poses are already established. Thus, in NeRF-related literature, Colmap is frequently employed to calculate camera-intrinsic and camera-extrinsic parameters. Some studies enhance camera positions using NeRF photometric loss [31,32]; however, this procedure necessitates lengthy training durations. In response to this issue, Instant-NGP has created a system that can rapidly train NeRF by utilizing multi-resolution hashing and the Compute Unified Device Architecture (CUDA) platform [33]. Several studies have specifically concentrated on constructing maps within SLAM systems [34,35], or merging NeRF with SLAM systems [36,37], showcasing commendable performance.

This research utilizes NeRF to reconstruct the scene in the VINS operating environment and generate virtual images according to the system's flight trajectory. It achieves this for extending the number of candidate frames and enhancing the probability of LCD.

3. Method

This paper proposes a BoW model-based LCD method using NeRF, with the overall method outlined in Figure 1. The red block at the top is the workflow of the VINS system, and the LCD method proposed in this paper is an important part of it. The general

workflow is as follows. First, feature points are extracted from keyframes; then, the camera-intrinsic and camera-extrinsic parameters of the original camera images are estimated using Colmap, and together with the images, they are put into Instant-NGP for scene reconstruction. After applying slight pose offsets, virtual viewpoints are selected for rendering corresponding virtual images. Subsequently, virtual images are filtered, and along with the original images, they form loop closure candidate frames. The cosine similarity between the loop closure candidate frames and the current frame is computed, and based on this, dynamic weights are assigned to calculate a comprehensive score to determine loop closure. This method mainly consists of three modules: key feature point extraction, virtual image construction and filtering based on NeRF, and loop closure determination based on cosine similarity calculation using the word frequency weight vector. Each of these modules will be detailed in subsequent sections.

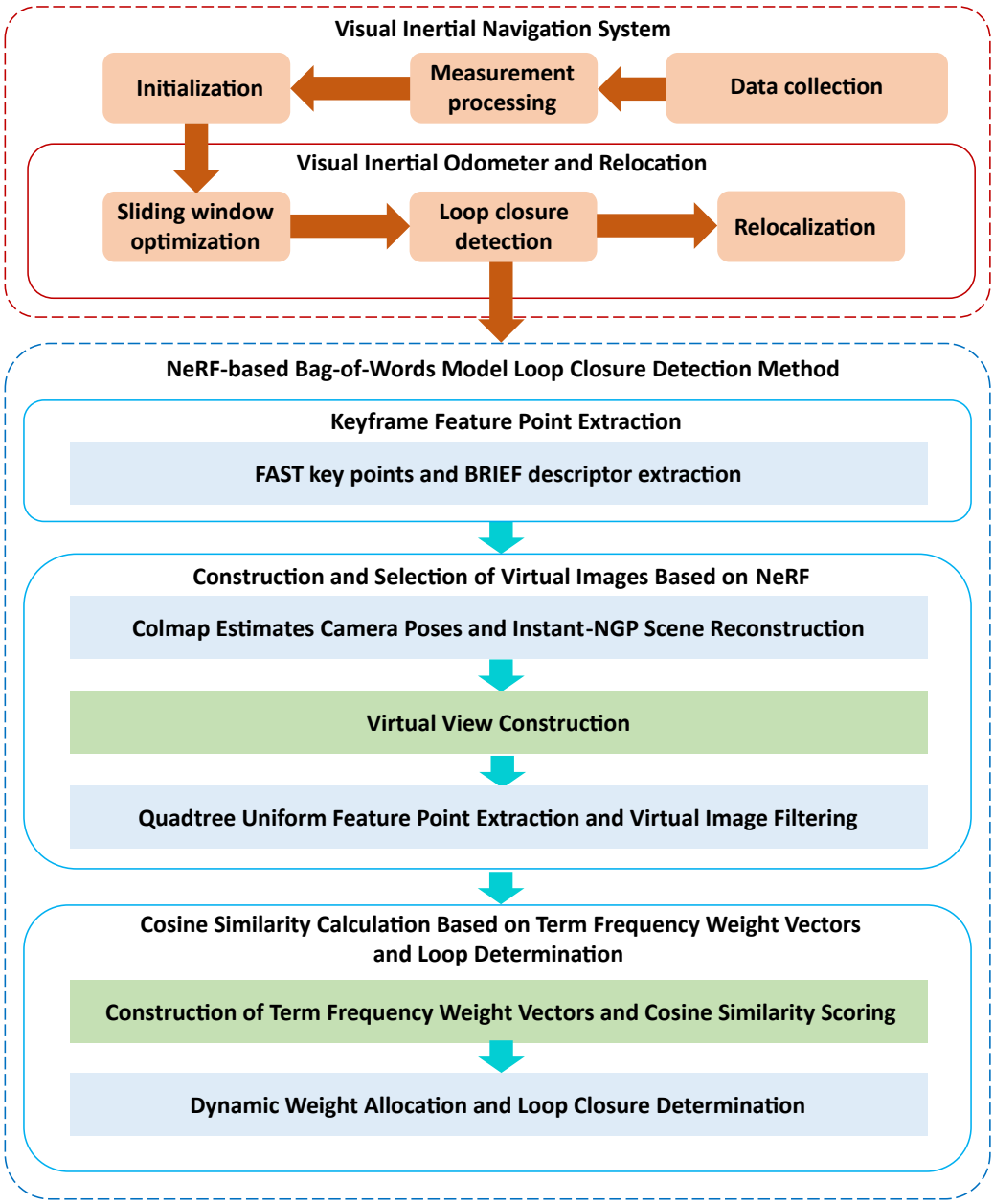

Figure 1. Framework of the BoW LCD method based on NeRF and its position in VINS.

3.1. Keyframe Feature Point Extraction

The first step in the BoW model for LCD is feature extraction. After downsampling the images returned by VIO, keyframes are selected and keypoints are extracted. FAST keypoints are detected by rapidly comparing the central pixel with its 16 surrounding pixels using Equation (1), as shown in Figure 2 [38]. Binary assignment and encoding are performed using Equations (2) and (3) to yield BRIEF descriptors [38]. These encoded descriptors along with keypoints are stored in a database for subsequent feature matching processes.

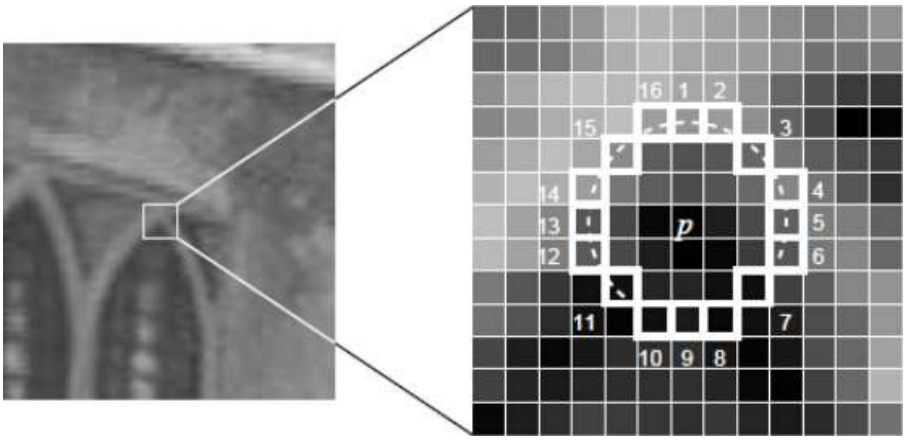

Figure 2. Positions of the central pixel and surrounding pixels.

$$N = \sum_{x \forall (circle(p))} |I(x) - I(p)| > \varepsilon_d, \tag{1}$$

$$\tau(p; x, y) = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{if } p(x) \geq p(y) \end{cases} \tag{2}$$

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x, y) \tag{3}$$

3.2. Construction and Selection of Virtual Images Based on NeRF

After extracting keyframes' feature points, the construction of virtual images can be initiated. The process is as follows.

3.2.1. Colmap Estimates Camera Poses and Instant-NGP Scene Reconstruction

Colmap is a structure-from-motion system that performs sparse reconstruction using scene information provided by images to obtain the camera-intrinsic and camera-extrinsic parameters, which are essential for generating virtual images. Colmap ultimately obtains the required camera-intrinsic and camera-extrinsic parameters by minimizing the bundle adjustment loss function, as Equation (4) [39]. In the equation, P_c represents the camera parameters, X_k denotes the point parameters, π is the projection function, ρ_j is the loss function, and x_j is the projected point.

$$E = \sum_j \rho_j \left(\left\| \pi(P_c, X_k) - x_j \right\|_2^2 \right), \tag{4}$$

NeRF is a novel scene reconstruction technique that learns the 3D representation of a scene from a collection of images with known camera viewpoints. Its input consists of spatial position (x, y, z) and viewing angles (θ, Φ). Its output includes the volume density and RGB values of each pixel under that pose. Pixel color is computed by integrating along sampled rays using volume rendering, as described in Equation (5) [11], where t_n is the near limit of the sampled ray, t_f is the far limit, $\sigma(t)$ is the volume density, T(t) denotes

the accumulated transmittance along the ray from t_n to t, c(r(t),d) is the expected color of camera ray r(t) = o + td with near and far bounds t_n and t_f, hld is the viewing direction. Instant-NGP utilizes multi-resolution hash encoding on the basis of the NeRF framework, resulting in a significant speed improvement.

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt \tag{5}$$

In practical experiments training NeRF, addressing the estimation of camera intrinsic and extrinsic parameters corresponding to images, two main approaches are currently employed: inertial visual bundle adjustment and visual bundle adjustment [40,41]. In this study, the results obtained from three camera parameter estimation schemes, including Colmap, are compared as detailed in Section 4.2. Each image's estimated camera-intrinsic and camera-extrinsic parameters, pixel dimensions of input images, scene size for rendering, and sharpness values are sent to Instant-NGP along with the original camera images to swiftly reconstruct the 3D scene.

3.2.2. Virtual View Construction

Upon completing Instant-NGP model training, the desired virtual camera pose can be selected, and its corresponding observed scene image can be rendered using the trained Instant-NGP model.

1. Coordinate Definition and Transformation
   Colmap employs a right-down-front coordinate system, whereas Instant-NGP uses a right-up-back coordinate system. Therefore, when converting the camera poses estimated by Colmap to the Instant-NGP coordinate system, only reversing the parameters of the Y-axis and Z-axis is required to perform the coordinate transformation.
2. Selection of Virtual View Poses
   To effectively synthesize virtual images during subsequent view rendering, it is necessary to select the position and orientation of virtual image views. Firstly, for position selection, in order to fully utilize the scene information, the sampling is conducted with each original image's capture position as the origin and a radius of 2 cm. Within this range, sampling is performed according to the following principles:

   - The distance between sampling points should not be less than a constant distance $\lambda_{\min}$ or greater than a constant distance $\lambda_{\max}$, to avoid the virtual view's sampling points being too dense or too sparse.
   - When the distance between adjacent original image capture positions is less than 2 cm, the generation of virtual views is abandoned to prevent overlap of adjacent sampling intervals when the vehicle moves slowly, resulting in misalignment of the virtual images.

   The nearest neighbor search is implemented using KD-trees. Firstly, k random three-dimensional points are generated within a spherical space of 2 cm around the sampling point, which is $p_1 = (x_1, y_1, z_1)$. The Euclidean distance between one of these random points $p_2 = (x_2, y_2, z_2)$ and the sampling point is calculated as follows:

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \tag{6}$$

   Then, all random points are sorted according to the x-coordinate dimension, and the median point is selected as the root node. The left subtree contains points with x-coordinates less than the median, while the right subtree contains points greater than the median. For each subtree, the next dimension (i.e., y, z coordinates) is recursively selected for partitioning until all dimensions are processed or the subtree contains only one point. At this point, the KD-tree construction is completed. Next, the search begins from the root node, and during the backtracking process, the distance between the target point and the node is calculated. The current nearest neighbor item is updated,

and distance constraints of $\lambda_{\min}$ and $\lambda_{\max}$ are added to the obtained point set, followed by pairwise distance checks, until all sampling points satisfy the conditions.
Next, it is necessary to determine the camera's pose. To enrich the scene information, a significant overlap field of view between the virtual view and the original image is required when selecting the pose. In this study, when selecting the camera pose, a small perturbation is added to each element of the rotation matrix R, where the perturbation amount falls within the range of $\left[-\frac{\theta}{2}, \frac{\theta}{2}\right]$, with θ being the maximum perturbation amplitude.

3. Virtual View Rendering
After the pose selection process, the chosen pose information can be input into the trained Instant-NGP model for rendering the virtual images. Figure 3 illustrates an example of virtual view generation, where the yellow box represents the selected virtual camera pose. In this study, rendering is conducted at half resolution (376 × 240), and then upsampled to the original size (752 × 480) using Fast Super-Resolution Convolutional Neural Network (FSRCNN) [42]. Rendering a single frame takes approximately 600 ms, achieving a balance between speed and quality.

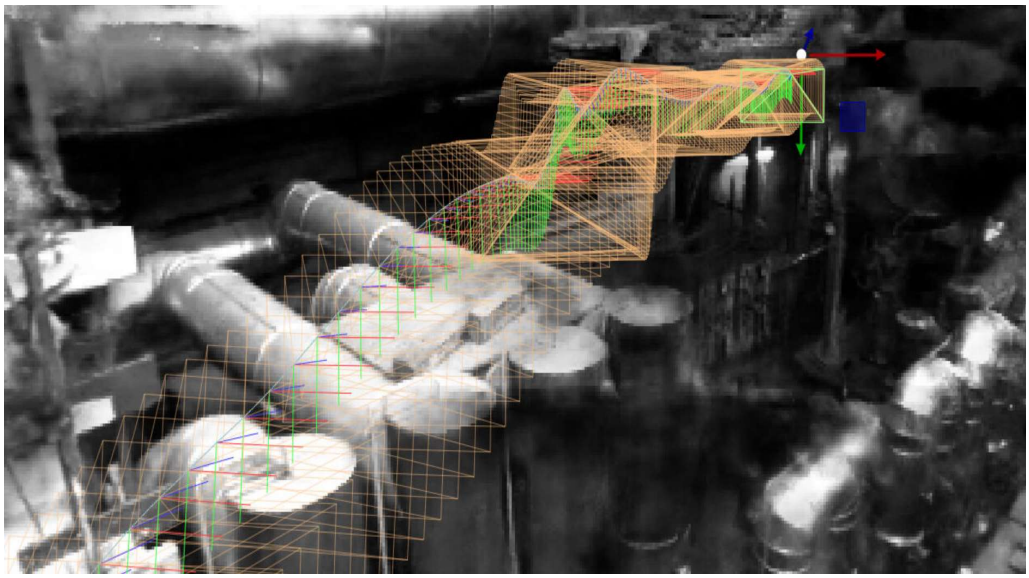

Figure 3. Example of Instant-NGP virtual view camera pose.

Then, the rendered virtual view images are synchronized in time with the corresponding original images from the dataset and added as new topics to the rosbag of the original dataset, awaiting processing.

3.2.3. Quadtree Uniform Feature Point Extraction and Virtual Image Filtering

After generating the virtual images, the system has to process several times more virtual images than the original ones. To alleviate the computational burden, the system needs to select the best candidate frames from the virtual images for each timestamp. To improve efficiency, the strategy of the feature point extraction is changed. The images are divided into four regions using a quadtree, as shown in Figure 4 [43]. Then, it is determined whether to continue dividing the regions based on whether the number of feature points in the divided region exceeds a threshold t. If the number is greater than t, further division is continued; otherwise, it is stopped.

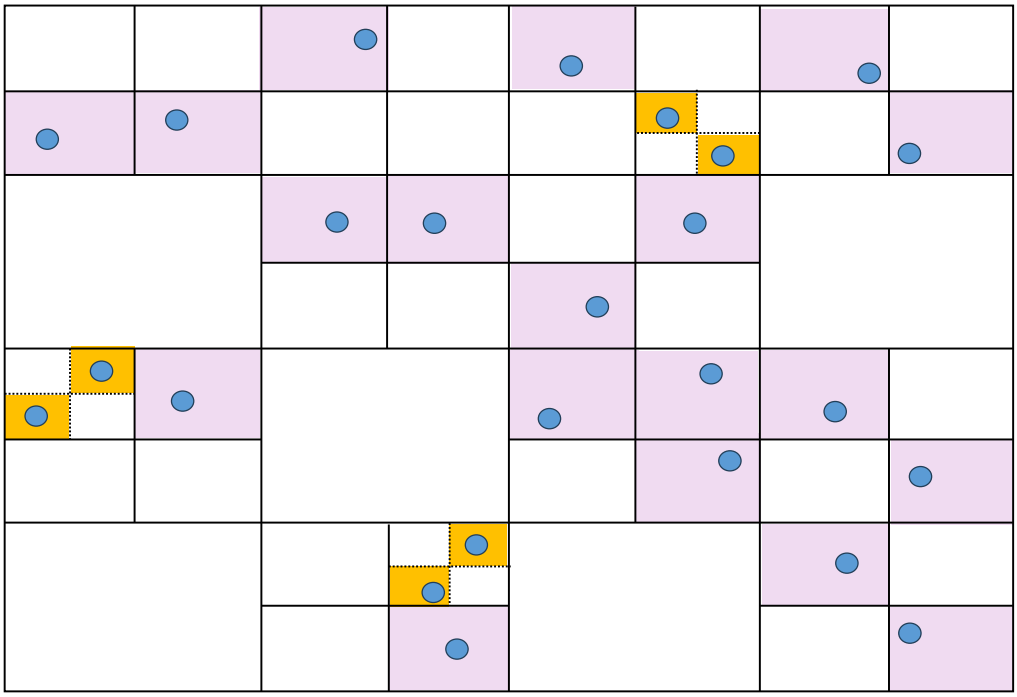

Figure 4. Quadtree uniform feature point extraction.

When the number of all extracted feature points exceeds the threshold T, all the division stops. The threshold T is set based on the average grayscale value within the grid. The calculation method is shown in the following formula [44]:

$$T = \sqrt{\frac{\sum (I(x,y) - \bar{I}(x,y))}{W \times H}}, \tag{7}$$

where $\bar{I}(x,y)$ is the mean grayscale value within the grid, the grid's width is W, and the height is H. Once all regions have stopped dividing, the similarity score between the n synthesized virtual images and their corresponding original images is calculated using vector norms [45]. The virtual image with the highest score becomes the final virtual candidate frame. Then, feature points are extracted using the method described in Section 3.1, awaiting subsequent computation.

3.3. Cosine Similarity Calculation Based on Term Frequency Weight Vectors and Loop Determination

After filtering the virtual images, term frequency weight vectors are constructed as descriptive vectors for each image, and cosine similarity calculations are performed. Loop closures are determined based on the comprehensive scores obtained from dynamic weight allocation.

3.3.1. Construction of Term Frequency Weight Vectors and Cosine Similarity Scoring

In BoW models, the Term Frequency–Inverse Document Frequency (TF-IDF) method is commonly used to construct term frequency vectors [44]. This method evaluates the weight of a word by combining term frequency (TF) and inverse document frequency (IDF). However, the expansion of virtual candidate frames introduces many "synonyms"—words similar to those in the original images—when describing the scene. The introduction of synonyms may cause mismatches and affect the position correction after LCD. Therefore, this study incorporates parameters to identify the similarity of "synonyms" when constructing BoW vectors.

After clustering feature points and descriptors using the K-means method, a dictionary is obtained, and BoW vectors can be used to represent the images. We then construct a BoW vector $S = [S_{\omega_1}, \ldots, S_{\omega_j}, \ldots, S_{\omega_k}]$ with a dimension of k, where S_{ω_j} is the frequency weight of the word ω_j. The calculation method is as follows:

$$S_{\omega_j} = S_{\omega_j}^{(TF-IDF)} \times \frac{1}{1 + \left\| I(\omega_j) \right\| \left\| \overline{S(\omega)} \right\| \cos\theta_j}, \tag{8}$$

where $S_{w_j}^{(TF-IDF)}$ is the TF-IDF score of the word w_j, calculated as shown in Equation (9). $F(w_j)_d$ is the ratio of the number of times the word w_j appears in the image to the total number of words in the image. *Sum* represents the total number of images in the database, and $P(w_j)_c$ represents the number of images in the database where the word w_j appears.

$$S_{w_j}^{(TF-IDF)} = F(w_j)_d \times \log(Sum/P(w_j)_c) \tag{9}$$

In Equation (8), $I(\omega_j)$ is the unit vector indicating the word ID. For example, when $I(\omega_1) = (0, 1, 0, \ldots, 0)$, it indicates that the word ID is 2. θ_j is the angle between the vector and the normalized vector $I(\omega_j)$. $\overline{S(\omega)}$ is the weighted expectation score vector of the k words that appear in the image, calculated as shown in Equation (10).

$$\overline{S(\omega)} = \sum_{j=1}^{k} \left(\frac{1}{1 - \frac{S_{\omega_j}^{(TF-IDF)}}{\sum_{j=1}^{k} S_{\omega_j}^{(TF-IDF)}}} \times I(\omega_j) \right) \tag{10}$$

This approach holds that words with similar meanings, which are in close proximity to the key vocabulary identified by TF-IDF, also possess a certain level of significance. Therefore, when introducing new words from the virtual image, it is necessary to assess whether these new words are semantically similar to the important vocabulary. This involves calculating the relevance of these new words to each word in the dictionary. This idea is realized by adding a fractional weighting factor after $S_{w_j}^{(TF-IDF)}$. In this weighting factor, $\overline{S(\omega)}$ calculates the proportion of each word's score in TF-IDF to the total score of all words. Subtracting this proportion from 1 and placing it in the denominator is performed to convert this weighting information into a number greater than 1. This is primarily to ensure significant differences in scores in subsequent calculations and to avoid all word weights being densely distributed in the interval [0,1]. Ultimately, we can consider that $\overline{S(\omega)}$ measures the "importance" of all words. $\left\| I(\omega_j) \right\| \left\| \overline{S(\omega)} \right\| \cos\theta_j$ then represents the similarity between word w_j and these important or unimportant words by computing the cosine value of two normalized vectors. "Synonymous" words similar to important words will receive a higher weighting ratio, while "synonymous" words similar to unimportant words may receive lower weights or even be ignored.

After constructing the BoW vectors, it is necessary to choose a similarity calculation method to assess the similarity between these BoW vectors. In BoW models, vector norms are commonly used to compute Euclidean distance or Manhattan distance to represent similarity. Their calculation methods are shown, respectively, in Equations (11) and (12), where $A = (A_1, A_2, \ldots A_k), B = (B_1, B_2, \ldots B_k)$ are two k-dimensional BoW vectors.

$$Euclidean: \quad d(A, B) = \sqrt{\sum_{i=1}^{k} (A_i - B_i)^2} \tag{11}$$

$$Manhattan: \quad d(A, B) = \sum_{i=1}^{k} |A_i - B_i| \tag{12}$$

The advantage of vector norms lies in its simple and fast calculation process, which meets the real-time requirements well. However, the drawback is its susceptibility to noise,

where fluctuations in the absolute values of each element in the vector may affect the similarity calculation results.

The cosine similarity used in our method mitigates the potential impact of noise through normalized vectors [45]. Additionally, in high-dimensional spaces like BoW vectors, Euclidean distances between vectors tend to be very close, which can render distance-based similarity calculation methods based on vector norms ineffective. Cosine similarity, on the other hand, focuses more on directional differences, allowing it to ignore differences in absolute values and providing a better assessment of similarity. The cosine similarity between BoW vectors can be calculated using Equation (13), where x_i, y_i represents the i-th element of the BoW vectors x and y.

$$sim(x,y) = \frac{x \cdot y}{\|x\|\|y\|} \cdot \frac{\sum_{i=1}^{k} x_i \times y_i}{\sqrt{\sum_{i=1}^{k} (x_i)^2} \times \sqrt{\sum_{i=1}^{k} (y_i)^2}} \tag{13}$$

3.3.2. Dynamic Weight Allocation and Loop Closure Determination

After calculating the cosine similarity between the candidate frames from the original images and the current frame, as well as between the candidate frames from the virtual images and the current frame at the same timestamp, dynamic weighting allocation is needed based on the scores. This involves assigning higher statistical weights to candidate frames that are more similar to the current frame. Ultimately, the comprehensive score is calculated based on the dynamically adjusted weights. And a loop is determined by comparing it with a predefined threshold. Initially, the corresponding weights for the two images need to be initialized as Equation (14), denoted as $\omega_{ori}, \omega_{vir}$ for the original and virtual images, respectively.

$$\omega_{ori} = \omega_{vir} = 0.5 \tag{14}$$

Then, the similarity scores of the two comparisons, S_{ori}, S_{vir}, representing the similarity scores between the original image and the current frame, and between the virtual image and the current frame, are compared. The following formula is used to update the two weights:

$$\begin{cases} \omega_{ori} = \omega_{ori} + \alpha \times (S_{ori} - S_{vir}) & \text{if} \quad S_{ori} > S_{vir} \\ \omega_{vir} = \omega_{ori} + \beta \times (S_{vir} - S_{ori}) & \text{if} \quad S_{ori} < S_{vir} \end{cases} \tag{15}$$

The parameters α, β are step length parameters, which can be manually adjusted by observing the dataset to verify the effect. In the experiments of this study, $\alpha = 1, \beta = 0.5$ are used for adjustment. After determining the weight distribution, the two scores are added together as the final comprehensive score, which is then compared with the threshold r. If the final comprehensive score is greater than the threshold r, the system is considered to have encountered a loop.

4. Experiment

The Euroc dataset is used to evaluate the effectiveness of the proposed method in this study [46]. A comparison is made between the method developed in this study and the conventional BoW model LCD method, with a focus on loop detection effectiveness and navigation localization accuracy.

4.1. Dataset and Server Information

The Euroc dataset is widely used in robot vision SLAM, specifically designed for micro UAVs. This dataset contains high-quality sensor data collected from real-world environments, including stereo camera images, IMU data, and ground truth information. The environments in the Euroc dataset range from industrial scenes to office environments, covering a variety of complex scenarios, making it an ideal choice for testing and validating

SLAM algorithms. By providing diverse scenes and precise sensor synchronization data, the dataset has played an important role in advancing research in UAV autonomous navigation and environmental perception. The available data in the dataset and the computer server information used in this study are shown in Table 1. In this study, the MH_01_easy scene from the Euroc dataset is used.

Table 1. Euroc dataset and server information.

Visual–Inertial Sensor Unit	Ground Truth	Calibration	Server Information
Stereo Image (Aptina MT9V034 shutter, WVGA monochrome, 2 × 20 FPS)	Vicon motion capture system	Camera intrinsics	GPU: GeForce RTX3090, 24 G
MEMS IMU (ADIS16448, 200 Hz)	Leica MS50 MultiStation	Camera-IMU extrinsics	CPU: AMD EPYC 7542 32-core processor, 681 G
		Spatio-temporally aligned ground truth	

4.2. Experiment on Loop Closure Detection

This study initially compared three methods for estimating camera poses, using Colmap, maplab for visual bundle adjustment, and inertial visual bundle adjustment through the combination of Vicon2gt with IMU data and OptiTrack. Then, the scene reconstruction is performed based on the estimation results, and the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) values of the reconstructed images are used to measure the reconstruction effectiveness. The experimental results are presented in Table 2. Through comparison, it is found that the reconstruction based on the estimation results from Colmap yielded better results. The visual effect of the reconstructed scene compared to the original images above can be more intuitively observed in Figure 5, which is consistent with the results in Table 2.

Table 2. PSNR and SSIM values of the reconstructed results of three pose estimation schemes.

	Maplab	Colmap	Vicon2gt
PSNR mean	6.6649	9.2680	8.8081
SSIM mean	0.0931	0.1789	0.1663

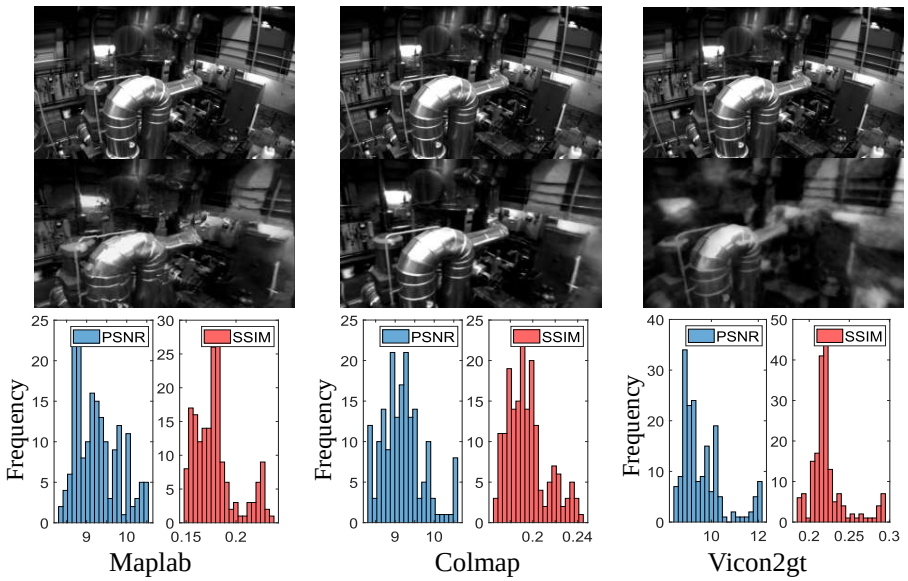

Figure 5. Comparison of reconstructed data of three pose estimation schemes.

After the reconstruction, virtual image rendering is conducted, and the feasibility of using the obtained virtual images as loop closure candidate frames is verified by feature matching with the original images. Through experiments, it is observed that the rendered virtual images and the original images can produce normal matches in the overlapping field of view area, as shown in Figure 6. This indicates that virtual images can effectively extract feature points, and the extracted feature points have the same image information representation effect as those extracted from real images. Therefore, expanding candidate frames with virtual images can provide richer scene information.

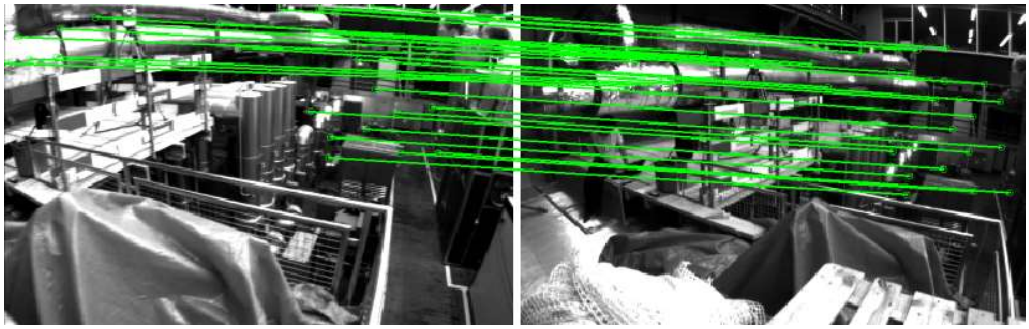

Figure 6. Feature matching effect between real image (**left**) and synthetic image (**right**).

Next, we will conduct experiments to evaluate the effectiveness of LCD. This study verifies the LCD performance by counting the number of loop closures detected and the accuracy of the detection. The BoW model-based LCD methods used in VINS-Mono and ORB-SLAM3 [47] are taken as the control for the experiments. The proposed NeRF-based BoW LCD method is applied to both systems, and its effectiveness is evaluated by comparing it with the original methods. In the BoW LCD methods of VINS-Mono and ORB-SLAM3, keyframes cannot be augmented, and their acquisition is solely based on camera captures. However, in the NeRF-based BoW LCD method, keyframes include both camera captures and virtual frames rendered by NeRF, significantly improving the success rate of matching with the current frame. The scene data from Euroc are used as the experimental input. We recorded the number of loop closures detected by both systems before and after applying the proposed NeRF-based BoW LCD method and determined the accuracy of these detected loop closures based on ground truth data. The experimental results are shown in Tables 3 and 4.

Table 3. Comparison of two BoW model LCD methods applied to VINS-Mono.

	Number of Detections (Times)	Accuracy Rate	Number of Additional Detection (Times)
BoW LCD method	121	100%	
NeRF-based BoW Model LCD Method	179	100%	58

Table 4. Comparison of two BoW model LCD methods applied to ORB-SLAM3.

	Number of Detections (Times)	Accuracy Rate	Number of Additional Detection (Times)
BoW LCD method	104	100%	
NeRF-based BoW Model LCD Method	171	100%	67

Figures 7 and 8 present the LCD results for the two approaches used in VINS-Mono and ORB-SLAM3. The vertical axis shows 1 for loop closure detected and 0 for not detected. The pink and yellow areas indicate the additional loop closure frames detected by the

NeRF-based BoW model with the LCD method, while the dark and purple areas represent loop closure frames detected by both methods.

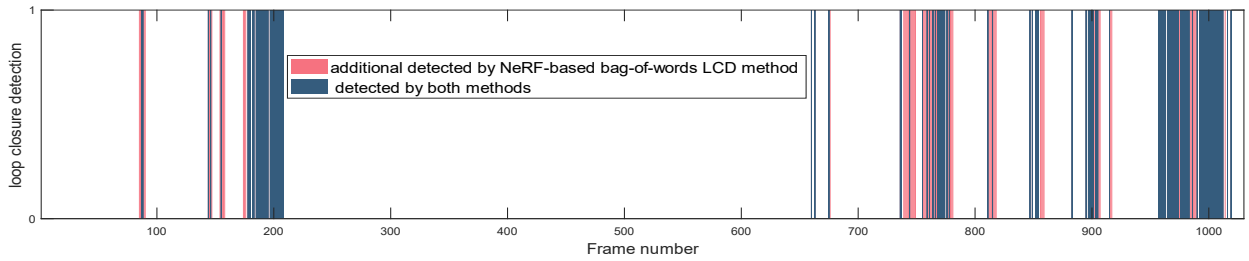

Figure 7. The loop closure frame detection results for the two approaches used in VINS-Mono.

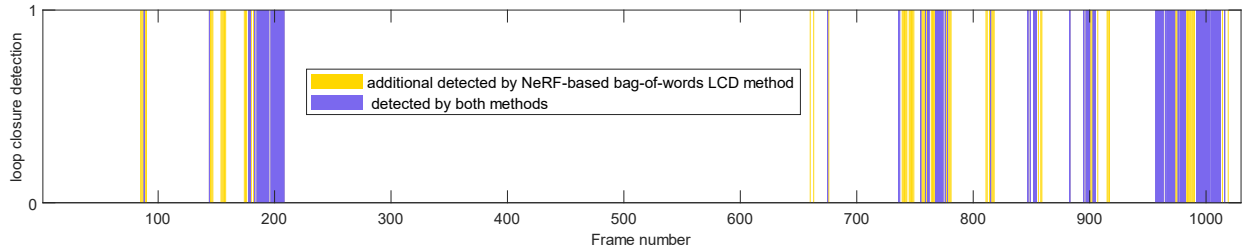

Figure 8. The loop closure frame detection results for the two approaches used in ORB-SLAM3.

The experiment demonstrates that by implementing the proposed NeRF-based BoW model LCD method, both systems can detect more accurate loop closures while maintaining detection accuracy. Taking one frame as an example for analysis, as shown in Figure 9, in the upper middle position, due to exposure reasons, few feature points can be extracted from this area in the current frame. However, on the right side, the virtual candidate frame, observed from a virtual viewpoint, has a weakened exposure situation, resulting in some more features extracted from the upper area of the image, and successfully achieved feature matching with the current frame exhibiting exposure conditions. It indicates that this method can increase the likelihood of detecting loop closures by increasing the number of features available for matching.

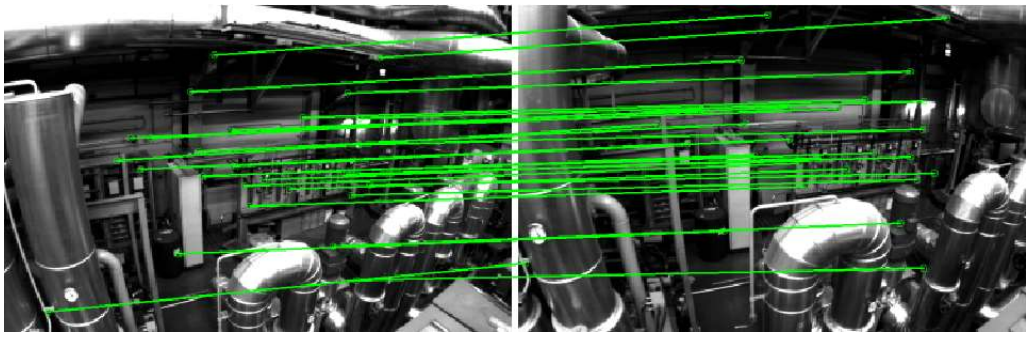

Figure 9. Example of additional loopback matching results.

4.3. Navigation and Localization Experiment

The main purpose of LCD is to correct localization errors. Therefore, this section will verify whether the proposed NeRF-based BoW model LCD method can effectively improve the navigation accuracy of the system. In this section, we apply the proposed NeRF-based BoW model LCD and the currently commonly used BoW model LCD to the VINS-Mono system and ORB-SLAM3 for visual–inertial navigation calculation. After obtaining the navigation trajectories, we compare them with the ground truth. First, we conduct a comparative experiment using the VINS-Mono system. The experimental trajectory image is shown in Figure 10.

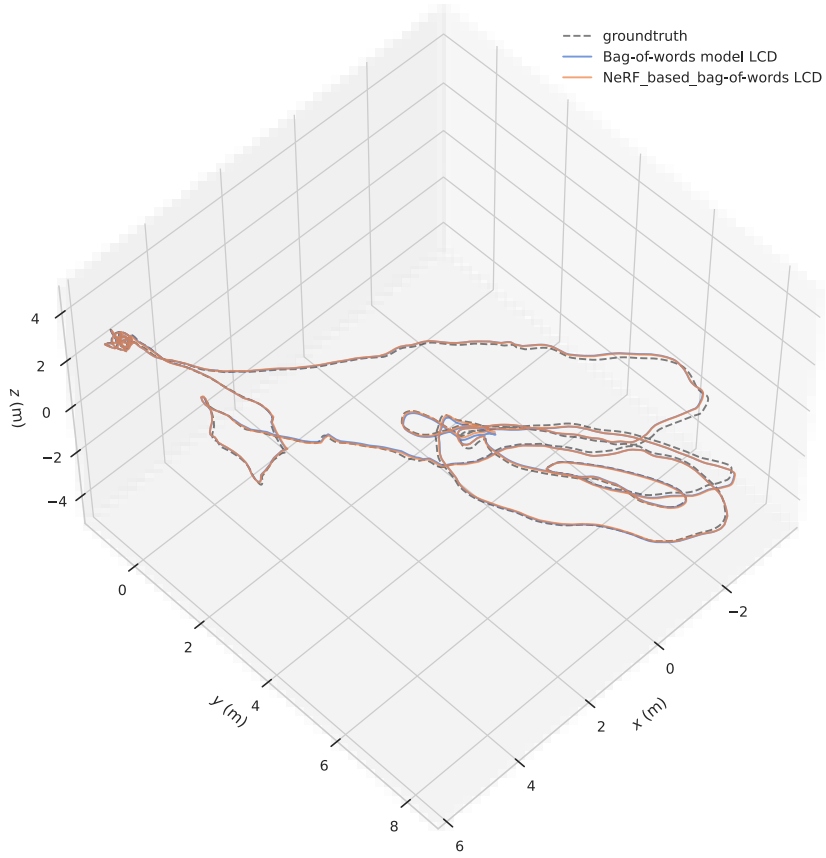

Figure 10. The ground truth and trajectory of two methods in VINS-Mono.

To comprehensively evaluate the accuracy of trajectory positioning, the Absolute Pose Error (APE) of the system is computed using the Evaluation of Odometry (EVO) tool as an indicator of navigation accuracy [48]. The experimental results are shown in Table 5. The distribution of APE error with image frame index is shown in Figure 11, and its specific statistical data are shown in Figures 12 and 13.

Table 5. Results of navigation experiment of two methods used in VINS-Mono.

	Max Error (m)	Min Error (m)	Mean of Error (m)	Rmse of Error (m)
VINS-Mono	0.79	0.01	0.15	0.18
NeRF+VINS-Mono	0.19	0.005	0.08	0.09

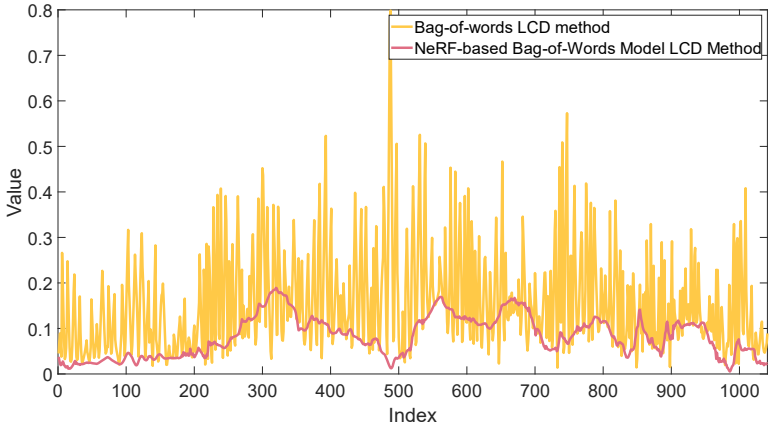

Figure 11. Statistical data of the APE with image frame index in VINS-Mono.

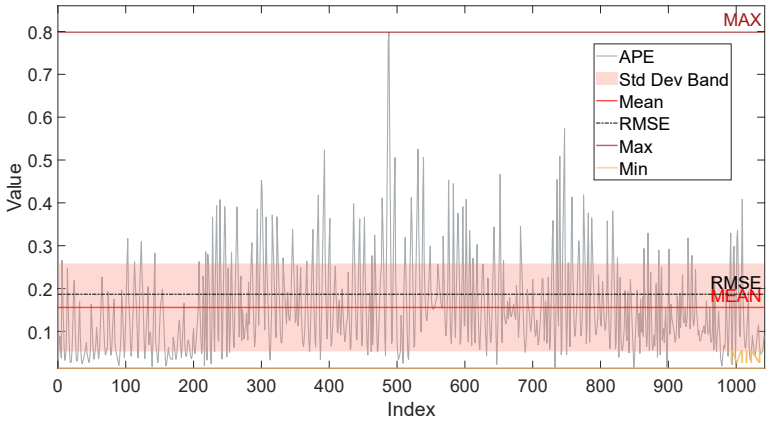

Figure 12. The APE statistics of BoW LCD method.

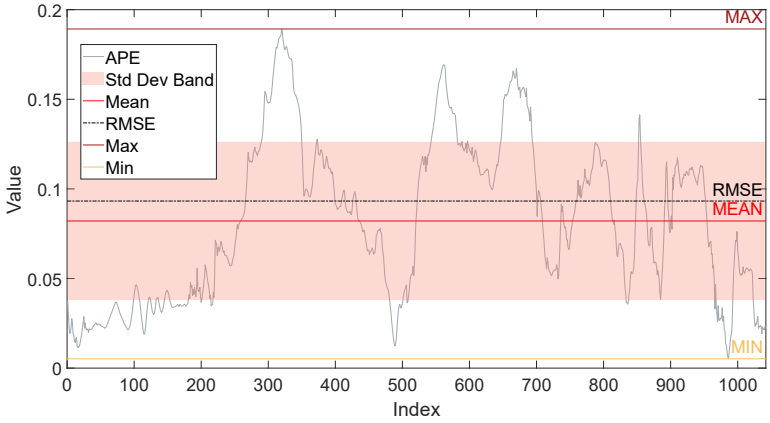

Figure 13. The APE statistics of NeRF-based BoW model LCD method.

The color distribution of APE along the trajectory in the VINS-Mono system is shown in Figure 14. Combined with the color bar, it can be observed that the NeRF-based BoW model LCD significantly improves the positioning accuracy of the system. The maximum trajectory error is reduced by 24%, the minimum trajectory error is reduced by 50%, the mean error is reduced by 53%, and the root mean square error is reduced by 50%. This indicates that the 58 additional loop closures detected by the NeRF-based BoW model LCD lead to more accurate position corrections by the system, resulting in a significant reduction in positioning error.

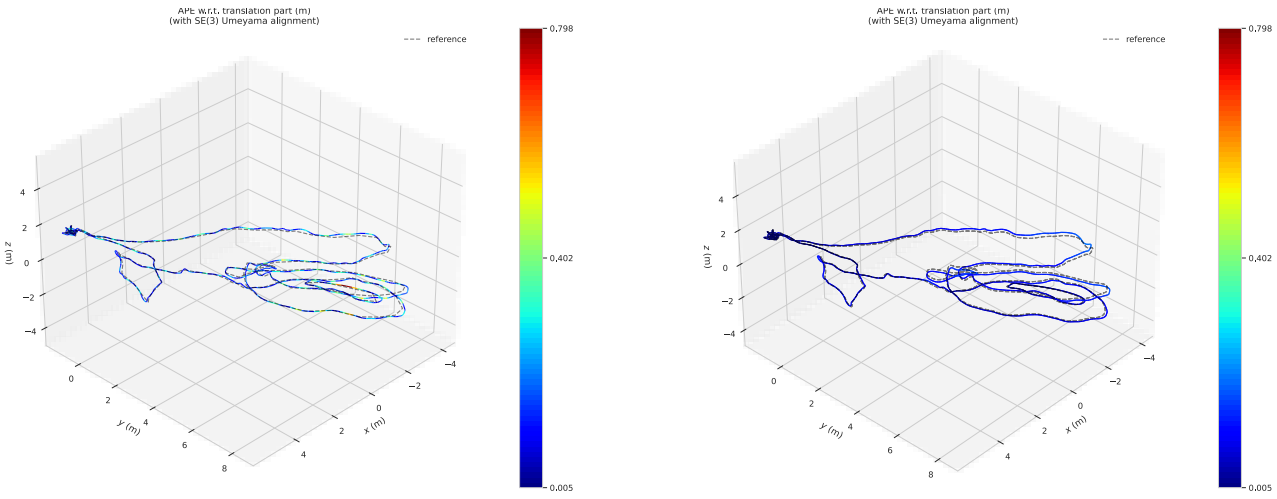

Figure 14. The distribution image of APE in the VINS-Mono system with the color of the trajectory.

The above experiment demonstrates that the proposed method can help the VINS-Mono system achieve better LCD and navigation performance. Next, the proposed method will be applied to the ORB-SLAM3 system. The experimental trajectory images are shown in Figure 15.

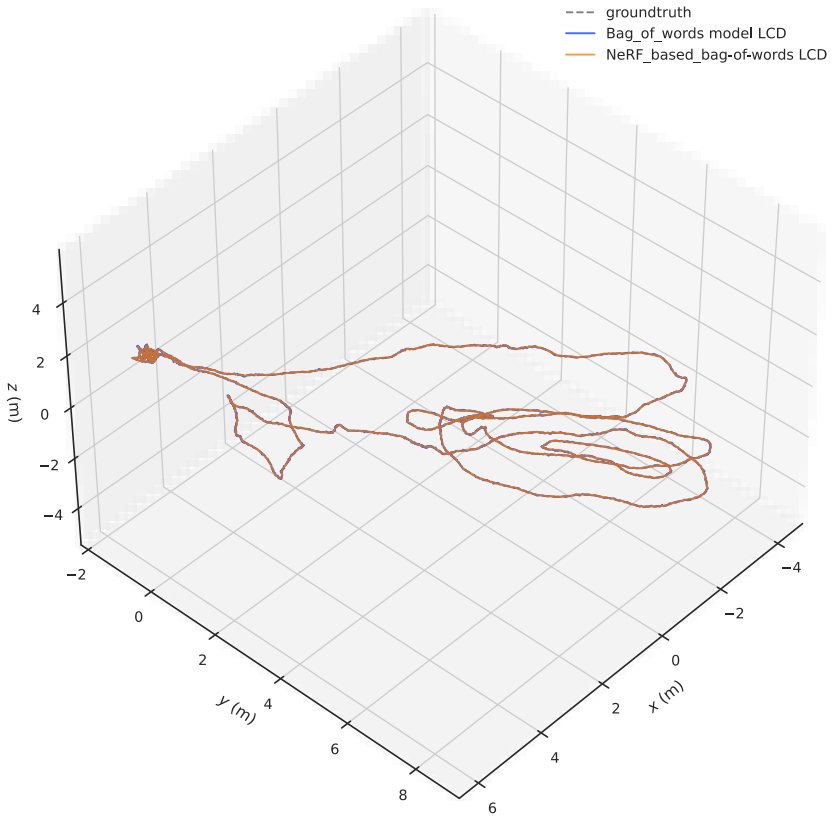

Figure 15. The ground truth and trajectory of two methods in ORB-SLAM3.

Using the EVO tool to calculate the system's APE, the experimental results are shown in Table 6. The APE data distribution over image frame numbers for both methods is shown in Figure 16, and the APE statistical data for the two methods are presented in Figures 17 and 18.

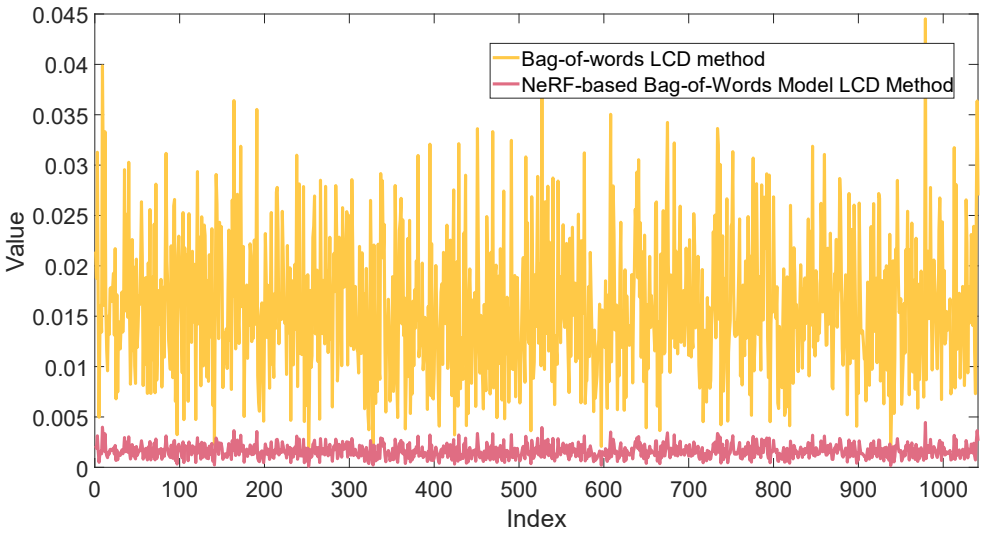

Figure 16. Statistical data of the APE with image frame index in ORB-SLAM3.

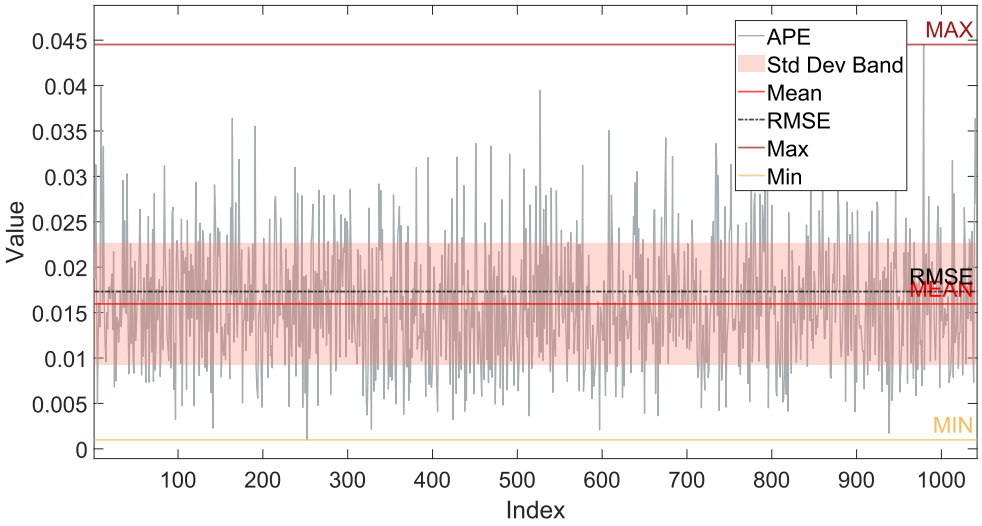

Figure 17. The APE statistics of BoW LCD method in ORB-SLAM3.

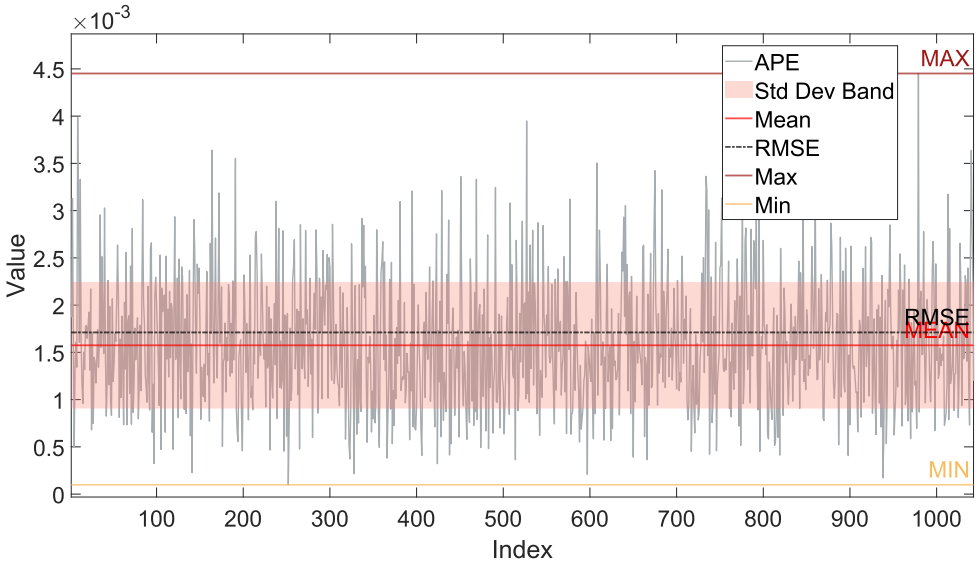

Figure 18. The APE statistics of NeRF-based BoW model LCD method in ORB-SLAM3.

Table 6. Results of navigation experiment of two methods used in ORB-SLAM3.

	Max Error (m)	Min Error (m)	Mean of Error (m)	Rmse of Error (m)
ORB-SLAM3	0.0445	0.00098	0.0157	0.0171
NeRF+ORB-SLAM3	0.0044	0.00009	0.0015	0.0017

The distribution image of APE in the ORB-SLAM3 system with the color of the track is shown in Figure 19. Combined with Colorbar, it can be found that the positioning accuracy of the ORB-SLAM3 system has been significantly improved by the LCD of the word bag model based on NeRF, and the maximum trajectory error has been reduced to 9% and the minimum trajectory error has been reduced to 9%. The mean error is reduced to 9.5%, and the root mean square error is reduced to 10%. This shows that 67 more loops detected by the NERF-based word bag model LCD result in more accurate position corrections and significantly reduced positioning errors.

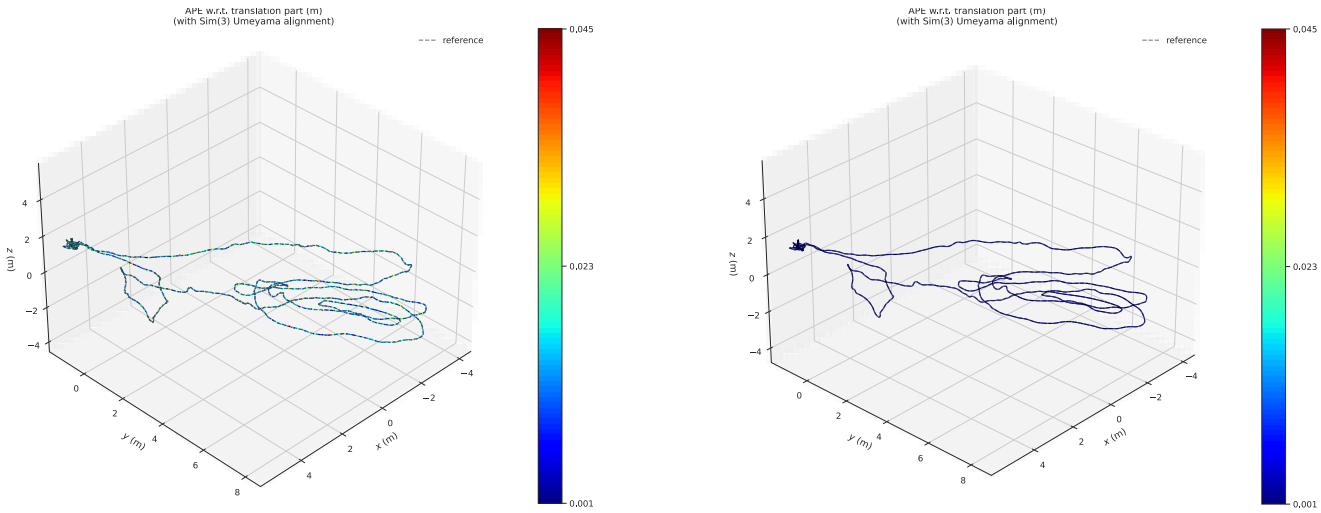

Figure 19. The distribution image of APE in the ORB-SLAM3 system with the color of the trajectory.

4.4. System Running Time Statistics and Algorithm Complexity Evaluation

Next, we will evaluate the system's runtime and the complexity of the algorithm. The method proposed in this paper is primarily applied to small UAVs, which require high real-time performance. Therefore, the computational complexity of the method is assessed from two perspectives.

First, in Section 4.4.1, we mainly evaluate the time complexity of the method, i.e., the time required for the method to run. This evaluation will provide a detailed analysis of the execution time of the algorithm with different input scales. By combining experimental data with theoretical analysis, we determine the time complexity of the method. The purpose of the time complexity evaluation is to ensure that the method can meet the real-time requirements in practical applications on small UAVs, guaranteeing that the system can respond and process data quickly during actual flight, thereby achieving stable navigation and positioning functions.

Secondly, in Section 4.4.2, we assess the space complexity of the method, i.e., the memory required during the operation of the method. The space complexity evaluation will consider the storage space needed by the algorithm for different input scales. By analyzing the space complexity, we can determine whether the method is feasible on resource-constrained small UAV platforms and optimize the method in subsequent research to reduce memory usage and improve overall system performance.

Through the evaluations of these two aspects, we can comprehensively understand the runtime efficiency and resource requirements of the proposed method in practical applications. This understanding provides a critical basis for subsequent optimization and improvement, ensuring the efficient and reliable application of the method on small UAVs.

4.4.1. Statistics on LCD Method Running Time

This experiment evaluates the time complexity of the proposed NeRF-based LCD method by comparing the time required for the VINS-Mono and ORB-SLAM3 systems to run the MH_01_easy scenario with both the proposed method and the original LCD methods. Since the ORB-SLAM3 graphical interface does not provide a reference for runtime, we used the <chrono> library for time measurement. Timing code was added to the LoopClosing function, the core loop detection function, to measure the total duration from the start to the end of running the MH_01_easy scenario. In contrast, the VINS-Mono graphical interface provides ROS time as a reference, which directly reflects the total duration from the start to the end of the system's run. The experimental results are shown in Table 7.

Table 7. System runtime statistics.

	BoW	NeRF+BoW
VINS-Mono	186.24 s	198.94 s
ORB-SLAM3	207.00 s	216.23 s

The results indicate that the proposed method is indeed more complex in terms of time complexity compared to the original method, requiring a longer processing time. This is expected, as the system needs to perform more comparisons and process more images. When running the VINS-Mono system, the NeRF-Based BoW LCD method required 12.7 s more than the original method, while ORB-SLAM3 required 9.23 s more. The time complexity of VINS-Mono and ORB-SLAM3 increased by 6.8% and 4.4%, respectively. Since this additional time is primarily spent on reading and filtering virtual images, it does not impact the real-time performance of the systems.

4.4.2. Statistics on Memory Required for LCD Operation

This experiment evaluates the spatial complexity of the proposed NeRF-based LCD method by comparing the Resident Set Size (RSS) of the VINS-Mono and ORB-SLAM3 systems when equipped with the proposed method versus their original LCD methods. RSS was chosen as the specific metric because it measures the total memory used by the process, including all shared libraries, providing an accurate assessment of the spatial complexity of the proposed method. Since both VINS-Mono and ORB-SLAM3 run on the Ubuntu 18.04 operating system, we used the system monitoring and process management tool Htop for the statistics. The results are shown in Table 8.

Table 8. RSS data statistics of two systems running with different LCD methods.

	BoW	NeRF + BoW
VINS-Mono	1,429,204 kbytes	1,434,624 kbytes
ORB-SLAM3	751,636 kbytes	771,626 kbytes

The experimental results show that when running the VINS-Mono system, the NeRF-Based BoW LCD method requires 5420 kbytes more memory than the original method, while ORB-SLAM3 requires 19,990 kbytes more. The spatial complexity of VINS-Mono and ORB-SLAM3 increased by 0.3% and 2.6%, respectively. The actual physical memory usage of both systems did not change significantly when equipped with either method.

4.5. Parametric Sensitivity Analysis

The effectiveness of the proposed method in this study involves a critical parameter, the LCD comprehensive score threshold r. The setting of this parameter directly affects the sensitivity and performance of LCD. Due to variations in images captured by cameras in different scenarios, the r value must be adjusted accordingly. Using the MH_01_easy scenario from the Euroc dataset as an example, this study conducted a sensitivity test on the r value to evaluate the robustness and effectiveness of LCD under varying r values. The experimental results shown in Figure 20 illustrate the sensitivity of LCD performance to changes in the r value.

The optimal composite score thresholds r for the VINS-Mono and ORB-SLAM3 systems in LCD were found to be 2.15 and 1.55, respectively. Considering experimental costs, this experiment set the minimum interval for the threshold r at 0.05. Experimental results indicate that when r is below its maximum value, the number of detected loop closures is significantly higher than the number of correct loop closures. This implies that while lowering the threshold increases the number of detected loop closures, it also increases the number of false matches. False matches can adversely affect the position correction after LCD, so it is essential to minimize their occurrence. Conversely, a higher threshold

reduces the number of detected loop closures due to the stringent conditions, which in turn decreases the frequency of position corrections, thereby impairing the system's ability to improve navigation accuracy through LCD.

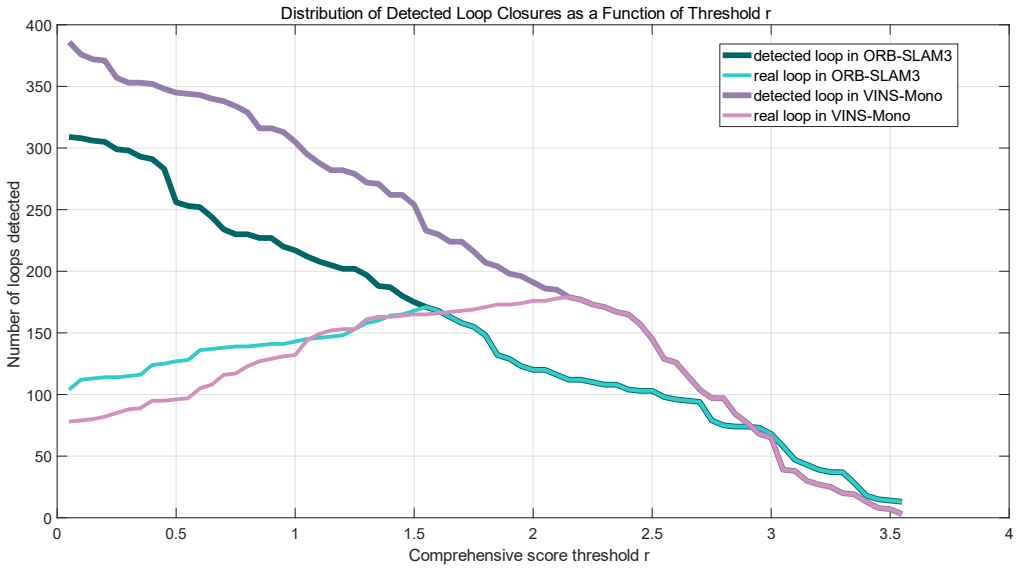

Figure 20. Distribution of detected loop closures as a function of threshold r.

5. Conclusions

This study proposed an LCD method based on NeRF and the BoW model, which features good real-time performance and high accuracy. It effectively reduces the difficulty of feature extraction and matches in the LCD process of the VINS system in dynamic scenes, weak texture environments, and lighting changes.

By incorporating NeRF, the LCD process gains richer observation perspectives and more feature information. A frequency-weighted vector based on similarity factors is designed to describe images in the candidate frame sequence composed of both virtual and original images. The method measures the correlation between vocabulary words, considers the importance of visually similar words introduced by virtual images, and formulates a corresponding dynamic weight allocation strategy to obtain comprehensive cosine similarity scores. With an improved LCD rate of 48% while maintaining accuracy, the mean positioning error of the VINS system is reduced by 53%. In the future, with computing power increases, the speed of NeRF operation will also significantly accelerate. We will continue to research integrating NeRF for online mapping and completing integration with other VINS systems to further improve LCD efficiency and navigation accuracy with real-time high-quality image rendering and scene reconstruction.

Nevertheless, there are three constraints to be tackled in this approach:

1. The efficiency of training the Instant-NGP model relies greatly on the quality of the data collected by the sensors, notwithstanding its ability to give detailed scene information and observer perspectives. The image quality of the model training is directly impacted by the shaking of the drone, leading to a loss in efficacy.
2. Conducting offline training for the Instant-NGP model necessitates extra storage capacity, and operating the VINS system and Instant-NGP also places demands on the system's operational memory.
3. The NeRF-based BoW model LCD method enhances the detection rate and accuracy of the VINS process. However, in environments with dynamic objects, the computational complexity of this method will increase, which may lead to a decrease in the success rate of detection and a longer response time. Additional trials are required to fine-tune the comprehensive scoring threshold used to identify loop closures, hence enhancing the stability and applicability of the LCD process to additional VINS systems.

In the future, with computing power increases, the speed of NeRF operation will also significantly accelerate. We will continue to research integrating NeRF for online mapping and completing integration with other VINS systems to further improve LCD efficiency and navigation accuracy with real-time high-quality image rendering and scene reconstruction.

Author Contributions: Conceptualization, Y.C. and H.Z.; Methodology, X.Z., Y.C. and H.Z.; Software, Y.C.; Validation, Y.C. and H.Z.; Formal analysis, X.Z., Y.C., Y.R., G.D. and H.Z.; Investigation, X.Z. and H.Z.; Resources, X.Z., Y.C., Y.R. and G.D.; Data curation, Y.C.; Writing—original draft, Y.C.; Writing—review & editing, X.Z., Y.R., G.D. and H.Z.; Visualization, Y.R. and G.D.; Supervision, X.Z., Y.R. and G.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Owing to the nature of the study, supporting data cannot be provided because study participants did not consent to the public sharing of their data.

Conflicts of Interest: Authors Yanchao Ren and Guodong Duan was employed by the company Hunan Vanguard Group Company Limited. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LCD	Loop Closure Detection
UAV	Unmanned Aerial Vehicle
VINS	Visual–Inertial Navigation System
BoW	Bag-of-Words
NeRF	Neural Radiance Fields
IMU	Inertial Measurement Unit
VIO	Visual Inertial Odometry
SLAM	Simultaneous Localization And Mapping
Instant-NGP	Instant Neural Graphics Primitives
CNN	Convolutional Neural Network
LRO	Local Relative Orientation
TSNE	T-distributed Stochastic Neighbor Embedding
VGG16	Visual Geometry Group 16
LBD	Line Band Descriptor
CUDA	Computer Unfied Device Architecture
FSRCNN	Fast Super-Resolution Convolutional Neural Network
TF-IDF	Term Frequency–Inverse Document Frequency
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
APE	Absolute Pose Error
EVO	Evaluation of Odometry
RSS	Resident Set Size

References

1. Li, X.; He, W.; Zhu, S.; Li, Y.; Xie, T. Survey of simultaneous localization and mapping based on environmental semantic information. *Chin. J. Eng.* **2021**, *43*, 754–767.
2. Xia, L.; Cui, J.; Shen, R.; Xu, X.; Gao, Y.; Li, X. A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420919185. [CrossRef]
3. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
4. Ding, W.; Xu, D.; Liu, X.-L.; Zhang, D.-P.; Chen, T. Review on Visual Odometry for Mobile Robots. *Zidonghua Xuebao/Acta Autom. Sin.* **2018**, *44*, 385–400.
5. Tsintotas, K.A.; Bampis, L.; Gasteratos, A. Modest-vocabulary loop-closure detection with incremental bag of tracked words. *Robot. Auton. Syst.* **2021**, *141*, 103782. [CrossRef]

6. Sivic, J.; Zisserman, A. Video Google: A text retrieval approach to object matching in videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; p. 1470.
7. Sun, H.; Wang, P.; Ni, C.; Li, J.M. Loop closure detection based on image semantic feature and BoW. *Multimed. Tools Appl.* **2022**, *83*, 36377–36398. [CrossRef]
8. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
9. Bansal, M.; Kumar, M.; Kumar, M. 2D object recognition: A comparative analysis of SIFT, SURF and ORB feature descriptors. *Multimed. Tools Appl.* **2021**, *80*, 18839–18857. [CrossRef]
10. Zheng, T.; Zhang, G.; Han, L.; Xu, L.; Fang, L. BuildingFusion: Semantic-Aware Structural Building-Scale 3D Reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2328–2345.
11. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *arXiv* **2020**, arXiv:2003.08934.
12. Guo, C.; Chen, X.; Song, J.; Hilliges, O. Human Performance Capture from Monocular Video in the Wild. In Proceedings of the 2021 International Conference on 3D Vision (3DV 2021), London, UK, 1–3 December 2021; pp. 889–898.
13. Niemeyer, M.; Geiger, A. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021), Nashville, TN, USA, 20–25 June 2021; pp. 11448–11459.
14. Kasten, Y.; Ofri, D.; Wang, O.; Dekel, T. Layered Neural Atlases for Consistent Video Editing. *ACM Trans. Graph.* **2021**, *40*, 210. [CrossRef]
15. Chung, C.-M.; Tseng, Y.-C.; Hsu, Y.-C.; Shi, X.-Q.; Hua, Y.-H.; Yeh, J.-F.; Chen, W.-C.; Chen, Y.-T.; Hsu, W.H. Orbeez-SLAM: A Real-Time Monocular Visual SLAM with ORB Features and NeRF-Realized Mapping. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA 2023), London, UK, 29 May–2 June 2023; pp. 9400–9406.
16. Rosinol, A.; Leonard, J.J.; Carlone, L. NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 3437–3444.
17. Mueller, T.; Evans, A.; Schied, C.; Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* **2022**, *41*, 102. [CrossRef]
18. Chen, Z.; Lam, O.; Jacobson, A.; Milford, M. Convolutional Neural Network-based Place Recognition. *arXiv* **2014**, arXiv:1411.1509.
19. Hou, Y.; Zhang, H.; Zhou, S. Convolutional Neural Network-Based Image Representation for Visual Loop Closure Detection. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2238–2245.
20. Ma, J.; Ye, X.; Zhou, H.; Mei, X.; Fan, F. Loop-Closure Detection Using Local Relative Orientation Matching. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 7896–7909. [CrossRef]
21. Zhan, H.; Zhu, Z.; Zhang, Y.; Guo, M.; Ding, G. Image sequence closed-loop detection based on residual network. *Laser Optoelectron. Prog.* **2021**, *58*, 315–323.
22. Sünderhauf, N.; Dayoub, F.; Shirazi, S.; Upcroft, B.; Milford, M. On the Performance of ConvNet Features for Place Recognition. *arXiv* **2015**, arXiv:1501.04158.
23. Luo, S.; Zhang, S. Convolutional neural network based loop detection algorithm. *Comput. Digit. Eng.* **2019**, *47*, 1020–1026.
24. Galvez-Lopez, D.; Tardos, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [CrossRef]
25. Labbe, M.; Michaud, F. Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation. *IEEE Trans. Robot.* **2013**, *29*, 734–745. [CrossRef]
26. Garcia-Fidalgo, E.; Ortiz, A. iBoW-LCD: An Appearance-Based Loop-Closure Detection Approach Using Incremental Bags of Binary Words. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3051–3057 [CrossRef]
27. Shi, J.; Meng, Q.; Dai, X. Visual SLAM loopback detection based on improved LBD and data-dependent metrics. *Laser Optoelectron. Prog.* **2021**, *58*, 291–299.
28. Martin-Brualla, R.; Radwan, N.; Sajjadi, M.S.M.; Barron, J.T.; Dosovitskiy, A.; Duckworth, D. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7206–7215.
29. Barron, J.T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; Srinivasan, P.P. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV 2021), Montreal, QC, Canada, 10–17 October 2021; pp. 5835–5844.
30. Barron, J.T.; Mildenhall, B.; Verbin, D.; Srinivasan, P.P.; Hedman, P. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022), New Orleans, LA, USA, 18–24 June 2022; pp. 5460–5469.
31. Jeong, Y.; Ahn, S.; Choy, C.; Anandkumar, A.; Cho, M.; Park, J. Self-Calibrating Neural Radiance Fields. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV 2021), Montreal, QC, Canada, 11–17 October 2021; pp. 5826–5834.
32. Lin, C.-H.; Ma, W.-C.; Torralba, A.; Lucey, S. BARF: Bundle-Adjusting Neural Radiance Fields. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV 2021), Virtual, 11–17 October 2021; pp. 5721–5731.

33. Adamkiewicz, M.; Chen, T.; Caccavale, A.; Gardner, R.; Culbertson, P.; Bohg, J.; Schwager, M. Vision-Only Robot Navigation in a Neural Radiance World. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4606–4613. [CrossRef]
34. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland, 26–29 October 2011; pp. 127–136.
35. Vespa, E.; Nikolov, N.; Grimm, M.; Nardi, L.; Kelly, P.H.J.; Leutenegger, S. Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1144–1151. [CrossRef]
36. Sucar, E.; Liu, S.; Ortiz, J.; Davison, A.J. iMAP: Implicit Mapping and Positioning in Real-Time. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021.
37. Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M.R.; Pollefeys, M. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
38. Rosten, E.; Porter, R.; Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 105–119. [CrossRef]
39. Schonberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
40. Cramariuc, A.; Bernreiter, L.; Tschopp, F.; Fehr, M.; Reijgwart, V.; Nieto, J.; Siegwart, R.; Cadena, C. maplab 2.0—A Modular and Multi-Modal Mapping Framework. *IEEE Robot. Autom. Lett.* **2023**, *8*, 520–527. [CrossRef]
41. Dasgupta, A.; Sharma, R.; Mishra, C.; Nagaraja, V.H. Machine Learning for Optical Motion Capture-Driven Musculoskeletal Modelling from Inertial Motion Capture Data. *Bioengineering* **2023**, *10*, 510. [CrossRef]
42. Dong, C.; Loy, C.C.; Tang, X. Accelerating the Super-Resolution Convolutional Neural Network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016. Available online: https://api.semanticscholar.org/CorpusID:13271756 (accessed on 17 August 2024).
43. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
44. Cummins, M.; Newman, P. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665. [CrossRef]
45. Jia, S.; Ma, L.; Tan, X.; Qin, D. Bag-of-Visual Words based Improved Image Retrieval Algorithm for Vision Indoor Positioning. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–4.
46. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
47. Campos, C.; Elvira, R.; Gómez Rodríguez, J.J.; Montiel, J.M. M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *arXiv* **2020**, arXiv:2007.11898.
48. Rebecq, H.; Horstschaefer, T.; Gallego, G.; Scaramuzza, D. EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time. *IEEE Robot. Autom. Lett.* **2017**, *2*, 593–600. [CrossRef]